ICTNET at Context Suggestion Track TREC 2012

Bingyang Liu[1,2], *Tong Wu*[1,2], *Xianghui Lin*[1,2], *Yanqin Zhong*[1,2] *Qian Liu*[1,2], *Yue Liu*[1], *Xueqi Cheng*[1]

1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190

2. Graduate School of Chinese Academy of Sciences, Beijing, 100190

{liubingyang,wutong,linxianghui,zhongyanqin,liuqian}@software.ict.ac.cn

{liuyue,cxq}@ict.ac.cn

Abstract

‘Yelp’ is used to collect the initial candidate web site list for every context. Then we use a spider to fetch contents of each candidate site. We also classify the candidates and examples into 6 classes and calculate preference of each profile to each class. The classes and preference are used for the ranking of the results. Finally we use several methods to filter and generate descriptions for every web site that is returned in the results.

1 Data Preparation

As there’s no data set provided, we need a straightforward method to limit the candidate websites for this task. Geographical position is what we used to filter out the millions possible places. Yelp and Google Maps are both tried and we decide to use Yelp. We use Yelp API to query a place S and get a list of places that are nearest to S. There’s a limitation of number of results returned by one yelp query, so we plot many places around S to get enough results from one single place. The keys returned by yelp API are as follows: latitude and longitude, name, category description, snippet text, homepage URL, reviews and business information of a place. The homepage URL is used by our spider to get the web pages on each place’s homepage.7265 websites are downloaded as our dataset.

2 Classification

In order to distinguish profiles from each other and keep diversity of the final results, a classification is needed.We separate the places into 6 classes represented by ABCDEF. The examples are classified manually. The places returned by Yelp API are classified according to their category description. A simple selected dictionary works well enough for this part, which saves our time on labeling training data. The classified examples are used to calculate the preference of each profile to each class.

3 Description

We put much emphasis on this part as we consider that the description is most important part for a person to judge whether this is an appropriate suggestion except the class of the place.The following 4 steps are used to filter and generate description for each place/website:

1. Description in html meta data(Identify);
2. Business information from Yelp API;
3. Dynamic generation using Background Model(Generation);
4. Snippet text from Yelp API.

These 4 steps are ordered by their qualifications of texts, not the difficulty of their implementations.

3.1 Identification

It is obvious that the first one html meta data description is good enough if the editor of the html is experienced and write it seriously. The problem is how can we distinguish the good written

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **NOV 2012**	2. REPORT TYPE	3. DATES COVERED **00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE **ICTNET at Context Suggestion Track TREC 2012**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Chinese Academy of Sciences,Institute of Computing Technology,Beijing, 100190,**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

Presented at the Twenty-First Text REtrieval Conference (TREC 2012) held in Gaithersburg, Maryland, November 6-9, 2012. The conference was co-sponsored by the National Institute of Standards and Technology (NIST) the Defense Advanced Research Projects Agency (DARPA) and the Advanced Research and Development Activity (ARDA). U.S. Government or Federal Rights License

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**3**	

descriptions from the bad ones. Here is what we do:

1. Extract meta data description part from html using regular expression.
2. Use GENIA Tagger to carry out a POS tagging on the description.
3. Classify the description into ‘Good’ and ‘Bad’ by SVM classifier. Features we used include if there exists a verb, the length and the value of number of words divided by number of punctuations.

We identified 2283 good descriptions among 7265 websites. We randomly evaluated 100 from the 2283 description and believe that they are written by good editors.

3.2 Generation

Step 3 is a dynamic generation based on Background Model proposed by Charles Clarke and William Song.For each suggested places, we are going to generate a short description within 512 characters. We suppose three collections C_R and C_{BG}. C_R is the collection of the reviews of a suggestion place. C_{BG} is the background of the suggested item’s class, In general, also including C_R .C is the combination of C_R and C_{BG}. The basic idea is to finding the sentences that are closest to the user interest and most different with the characteristics of the big class as description.

The detail process is as follows:

i. Define sentence s€C , define term€s

ii. For collections CR and CBG count the number of term in C, using frequency representing the probability Prob(term).

iii. For each term in CR, calculate the KL distance of term and CR. For each term in CBG, calculate the KL distance of term and CBG. Score(term)= KL(term||CR)-∂KL(term||CBG) , ∂ is the experience value.

iv. For each s€CR calculate the score of s, score(s) = $\sum \ score(term)$ s€CR，term€s

v. Sort the sentences and generate the description by scores.

4 Ranking

We only consider geographical information in RUN1 and take it as a baseline, while fitness, preference, diversity and some other filters are all evaluated to give a refined ranking result in RUN2. The description part are the same in RUN1 and RUN2.

The score of each place is calculated as follows:

Score = F(itness) + P(reference) + D(iversity) (1)

F = 0.1*Time + 0.32*distance
+ 0.12*reviews + 0.1*(IF Exist Homepage) (2)

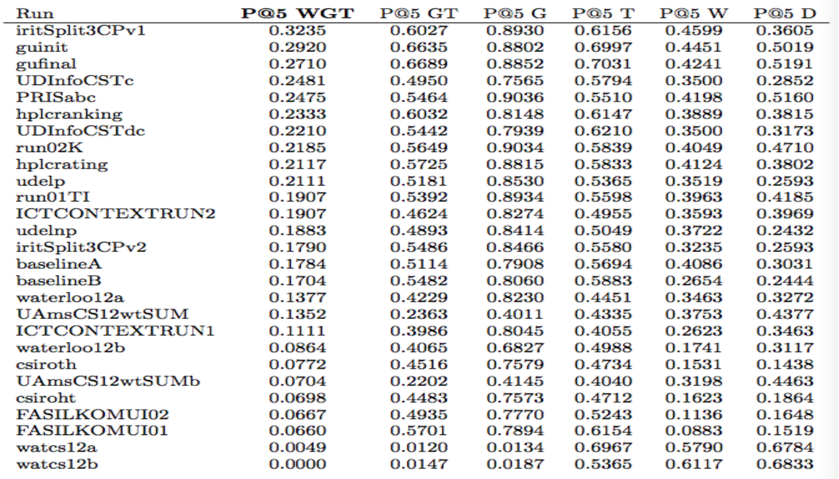
P = 0.27*(Preference for Each Class in examples) (3)

D = 0.09*(Entropy of the category discriptions) (4)

We normalize each factor and the parameters are selected using feedback method.

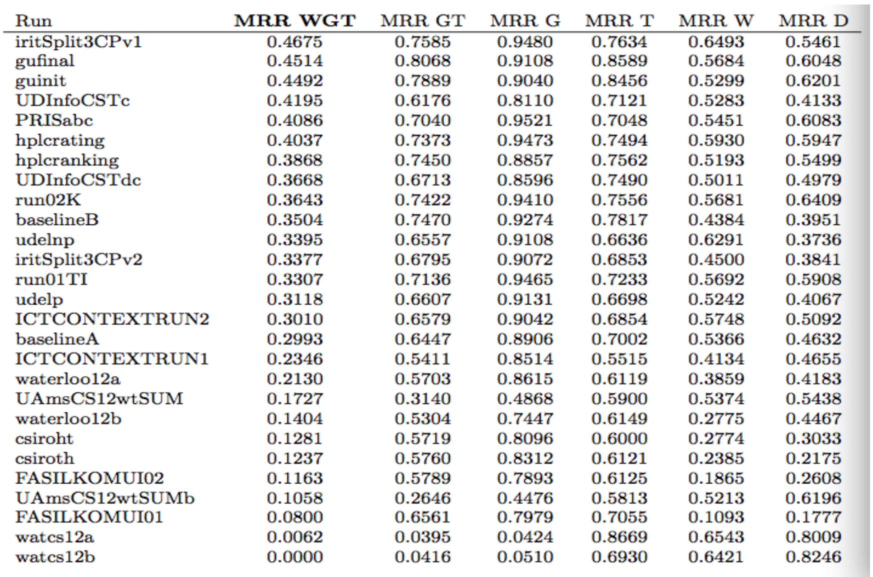
5 Result

We submitted two runs: ICTCONTEXTRUN1 and ICTCONTEXTRUN2. The results are as follows.



Run	P@5 WGT	P@5 GT	P@5 G	P@5 T	P@5 W	P@5 D
iritSplit3CPv1	0.3235	0.6027	0.8930	0.6156	0.4599	0.3605
guinit	0.2920	0.6635	0.8802	0.6997	0.4451	0.5019
gufinal	0.2710	0.6689	0.8852	0.7031	0.4241	0.5191
UDInfoCSTc	0.2481	0.4950	0.7565	0.5794	0.3500	0.2852
PRISabc	0.2475	0.5464	0.9036	0.5510	0.4198	0.5160
hplcranking	0.2333	0.6032	0.8148	0.6147	0.3889	0.3815
UDInfoCSTdc	0.2210	0.5442	0.7939	0.6210	0.3500	0.3173
run02K	0.2185	0.5649	0.9034	0.5839	0.4049	0.4710
hplcrating	0.2117	0.5725	0.8815	0.5833	0.4124	0.3802
udelp	0.2111	0.5181	0.8530	0.5365	0.3519	0.2593
run01TI	0.1907	0.5392	0.8934	0.5598	0.3963	0.4185
ICTCONTEXTRUN2	0.1907	0.4624	0.8274	0.4955	0.3593	0.3969
udelnp	0.1883	0.4893	0.8414	0.5049	0.3722	0.2432
iritSplit3CPv2	0.1790	0.5486	0.8466	0.5580	0.3235	0.2593
baselineA	0.1784	0.5114	0.7908	0.5694	0.4086	0.3031
baselineB	0.1704	0.5482	0.8060	0.5883	0.2654	0.2444
waterloo12a	0.1377	0.4229	0.8230	0.4451	0.3463	0.3272
UAmsCS12wtSUM	0.1352	0.2363	0.4011	0.4335	0.3753	0.4377
ICTCONTEXTRUN1	0.1111	0.3986	0.8045	0.4055	0.2623	0.3463
waterloo12b	0.0864	0.4065	0.6827	0.4988	0.1741	0.3117
csiroth	0.0772	0.4516	0.7579	0.4734	0.1531	0.1438
UAmsCS12wtSUMb	0.0704	0.2202	0.4145	0.4040	0.3198	0.4463
csiroht	0.0698	0.4483	0.7573	0.4712	0.1623	0.1864
FASILKOMUI02	0.0667	0.4935	0.7770	0.5243	0.1136	0.1648
FASILKOMUI01	0.0660	0.5701	0.7894	0.6154	0.0883	0.1519
watcs12a	0.0049	0.0120	0.0134	0.6967	0.5790	0.6784
watcs12b	0.0000	0.0147	0.0187	0.5365	0.6117	0.6833

Table 1: All 6 P@5 measures sorted by WGT.



Run	MRR WGT	MRR GT	MRR G	MRR T	MRR W	MRR D
iritSplit3CPv1	0.4675	0.7585	0.9480	0.7634	0.6493	0.5461
gufinal	0.4514	0.8068	0.9108	0.8589	0.5684	0.6048
guinit	0.4492	0.7889	0.9040	0.8456	0.5299	0.6201
UDInfoCSTc	0.4195	0.6176	0.8110	0.7121	0.5283	0.4133
PRISabc	0.4086	0.7040	0.9521	0.7048	0.5451	0.6083
hplcrating	0.4037	0.7373	0.9473	0.7494	0.5930	0.5947
hplcranking	0.3868	0.7450	0.8857	0.7562	0.5193	0.5499
UDInfoCSTdc	0.3668	0.6713	0.8596	0.7490	0.5011	0.4979
run02K	0.3643	0.7422	0.9410	0.7556	0.5681	0.6409
baselineB	0.3504	0.7470	0.9274	0.7817	0.4384	0.3951
udelnp	0.3395	0.6557	0.9108	0.6636	0.6291	0.3736
iritSplit3CPv2	0.3377	0.6795	0.9072	0.6853	0.4500	0.3841
run01TI	0.3307	0.7136	0.9465	0.7233	0.5692	0.5908
udelp	0.3118	0.6607	0.9131	0.6698	0.5242	0.4067
ICTCONTEXTRUN2	0.3010	0.6579	0.9042	0.6854	0.5748	0.5092
baselineA	0.2993	0.6447	0.8906	0.7002	0.5366	0.4632
ICTCONTEXTRUN1	0.2346	0.5411	0.8514	0.5515	0.4134	0.4655
waterloo12a	0.2130	0.5703	0.8615	0.6119	0.3859	0.4183
UAmsCS12wtSUM	0.1727	0.3140	0.4868	0.5900	0.5374	0.5438
waterloo12b	0.1404	0.5304	0.7447	0.6149	0.2775	0.4467
csiroht	0.1281	0.5719	0.8096	0.6000	0.2774	0.3033
csiroth	0.1237	0.5760	0.8312	0.6121	0.2385	0.2175
FASILKOMUI02	0.1163	0.5789	0.7893	0.6125	0.1865	0.2608
UAmsCS12wtSUMb	0.1058	0.2646	0.4476	0.5813	0.5213	0.6196
FASILKOMUI01	0.0800	0.6561	0.7979	0.7055	0.1093	0.1777
watcs12a	0.0062	0.0395	0.0424	0.8669	0.6543	0.8009
watcs12b	0.0000	0.0416	0.0510	0.6930	0.6421	0.8246

Table 2: All 6 MRR measures sorted by WGT.

The ICTCONTEXTRUN1 is under both baselineA and baselineB in both tables. We take ICTCONTEXTRUN1 as our baseline that contains all the raw results we fetched.

6 Acknowledgements

The context suggestion track is the first year and no one in this group really know what results can we get and how can we get them. We spent a lot of time grabbing the data, not to speak of using a lot of proxies to bypass the strong regulations of Yelp's firewall. Not all we planned are implemented finally subject to the limited time. But these written in this report are indeed accomplished and worked not badly. We hope a dataset can be provided in the next year. Thanks a lot to the hard working members of our group and to the TREC committees. This work is sponsored by NSF of China Grants No. 60933005, and by 863 Program of China Grants No. 2012AA011003.